

Operating PostgreSQL at Scale

Lessons from Hundreds of Instances
in Regulated Private Clouds

Swiss PGday 2025-06-12
Aarno Aukia

Agenda

- About
- History
- Requirements
- Challenges
- Outlook
- How are we using it today
- Customer examples
- (Live demo?)
- Q&A

About

Aarno Aukia

Co-founder & Partner

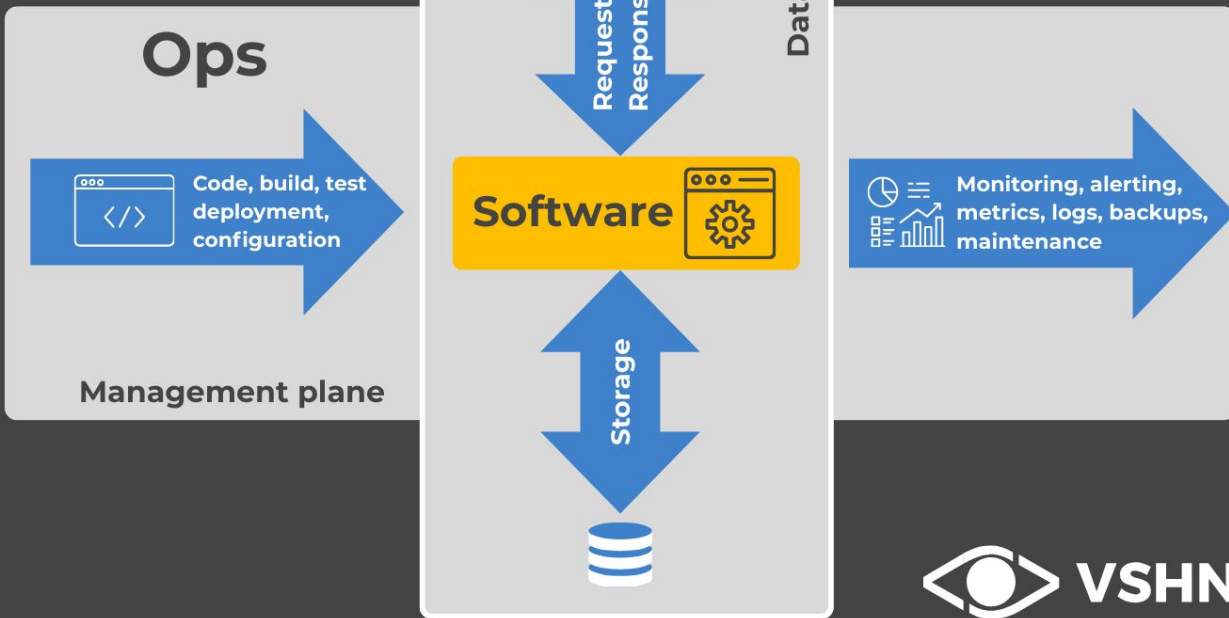
VSHN - The DevOps Company

VSHN automates and manages cloud and on-premises application operations, providing 24/7 support so developers and SaaS providers can focus on building since 2014.

Offices in Zürich, Switzerland, and Vancouver, Canada.



DevOps



DevOps Report 2025

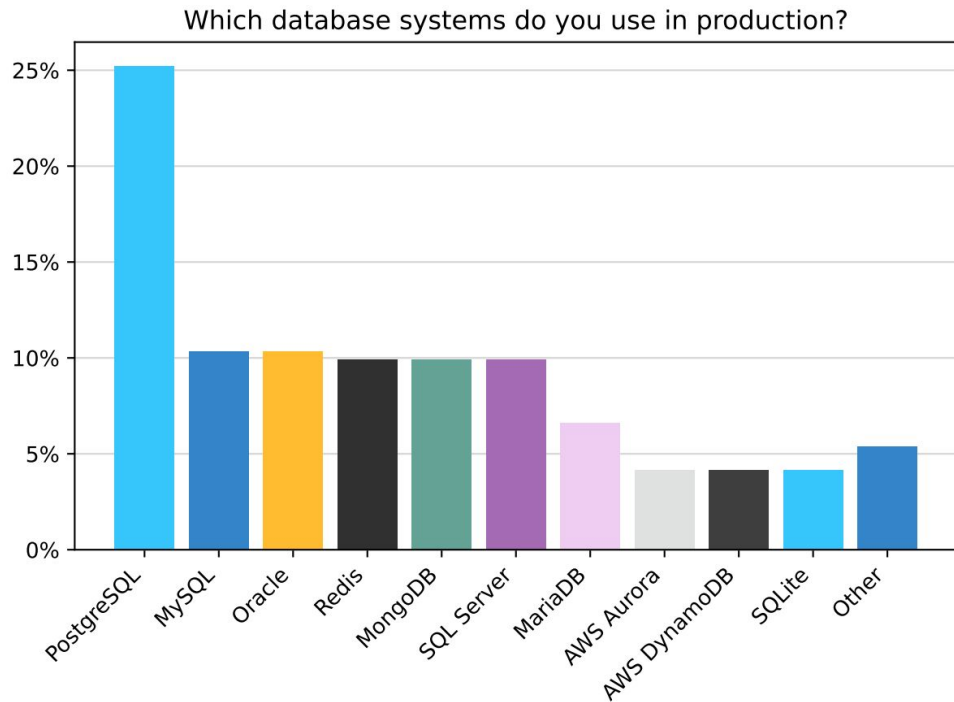
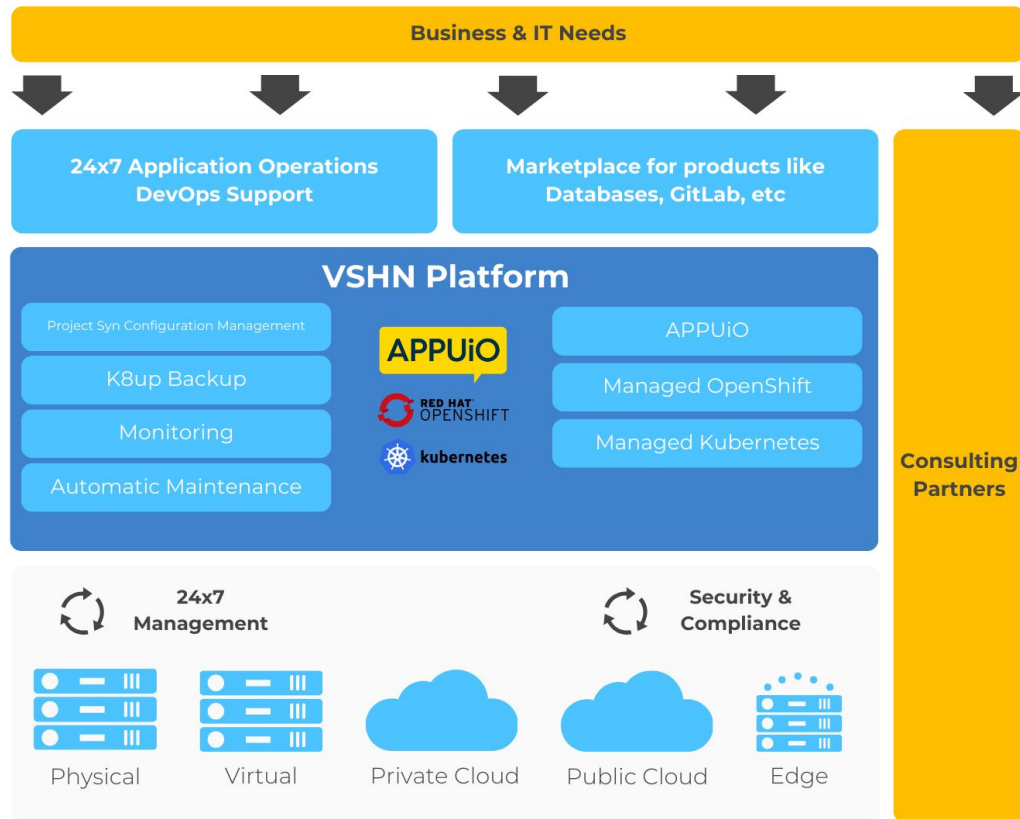


Figure 18. Database Services in 2025



<https://www.vshn.ch/devops-report-2025/>

VSHN - The 24x7 Platform Engineering Company



History 2014-2022

- Puppet & Ansible managed Linux Servers since 2014
 - manual script/terraform-based new server provisioning
 - automated backups, monitoring, maintenance, config management processes
 - no elasticity, auto scaling, etc
 - Very few customers are able to self-service

Needs 2022

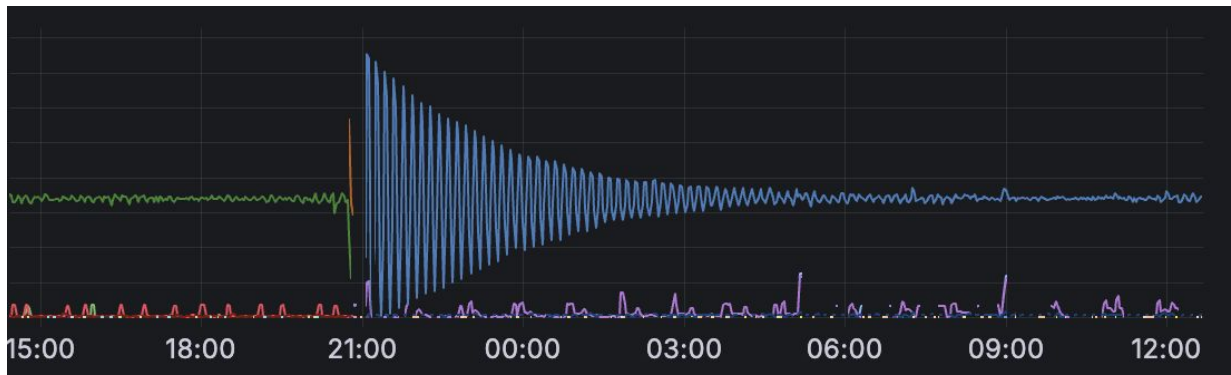
- Complete self-service
- fully automated monitoring, backup, maintenance
- Needs to work on-prem air gapped, private cloud, sovereign cloud, etc
- Using Kubernetes instead of VMs as infrastructure abstraction

Evaluation 2022

Requirements	Percona	Stackgres	CloudnativePG	Zalando	Crunchydata	Bitnami Helmchart
Superuser Access	✓	✓	✓	✓	✓	✓
Database Management [1]	✗	✗	✗	✓	✓	✗
User Management [1]	✗	✗	✗	✓	✓	✓
Service Metrics	✗	✓	✓	✗	✓	✓
Maintenance Schedule	✓	✓	✗	✗	✗	✗
Backup Schedule	✓	✓	✓	✓	✓	✗
Self Service Backup Restore	✓	✓	✓	✓	✓	✗
Encryption at rest [2]	✓	✓	✓	✓	✓	✓
Connection limit of 25/Cb Memory [3]	✓	✓	✓	✗	✓	✓
Extension Management [1]	✗	✓	✗	✓	✗	✓
Custom PostgreSQL Settings	✓	✓	✓	✗	✓	✓
In-Place Major Upgrade	✓	✓	✗	✗	✗	✗
License	Apache License 2.0	AGPL-3.0	Apache License 2.0	MIT	Apache License 2.0	Apache License 2.0
Upstream Support	Some support available not clear if for operators	✓	✓ (via EDB)	✗	✓	✗

Challenges 2022-2025

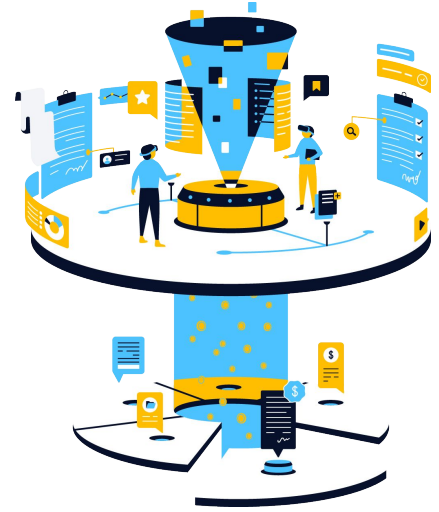
- StackGres features lacked production-grade reliability at scale.
- Operator performance degraded with many instances.
- Kubernetes control plane was overloaded due to reconciliation storms.
- Lack of commercial partnership/support became a blocker.



Outlook 2025+

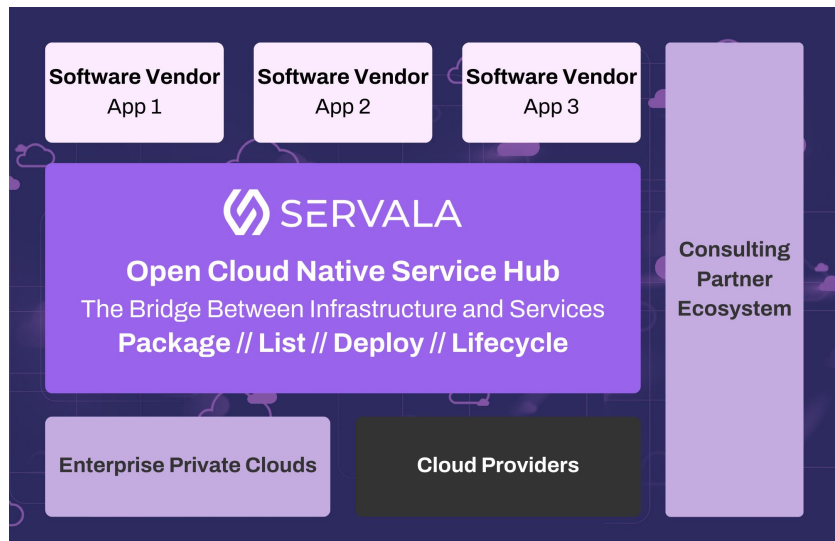
- Crunchy Data was acquired by Snowflake in June 2025, we have customers with similar issues like we did with Stackgres
- We're migrating to CloudNativePG because they made great progress in recent years and offer a stable, production-grade operator.
- EDB partnership considered a bonus.

So, how is this used?



Servala by VSHN

- Open-source “App Store” for managed services on Kubernetes and OpenShift
- runs air-gapped/on-prem, private cloud, sovereign cloud, public cloud
- 99.99% SLA, 24x7 Support, ISO 27001, ISAE 3402, SOC 2, BSI C5 available from VSHN



ISVs



Red Hat
Container Platform,
Developer Experience, AI



IBM
Watsonx Data & AI



GitLab
SCM, CI/CD, DevSecOps



PostgreSQL
Relational Database



Isovalent
Network Security &
Observability



Keycloak
IAM



Forgejo
Source Code Management



MariaDB
Relational Database



Nextcloud
Files & Groupware



Kafka
Event Streaming



Stardog
Knowledge Graph



Redis/Valkey
NoSQL Database



Airlock
Web Application Firewall



Grafana
Monitoring, Analytics, Tracing



Couchbase
NoSQL Database



EnterpriseDB
Relational Database



NGINX
Web server



Harbor
Enterprise Container Registry



MongoDB
NoSQL Database



MySQL
Relational Database



Odoo
ERP



RabbitMQ
Message Broker

PostgreSQL

[Website](#)

The World's Most Advanced Open Source Relational Database

About PostgreSQL

- AT&T
- Checksum
- Extensibility
- Foreign Data Wrappers
- InnoDB
- InnoDB
- Enterprise Private Cloud

External Links

- Amazon Aurora
- Microsoft

Related Articles

- PostgreSQL vs Oracle

Overview

PostgreSQL, a powerful, open source relational database system that works and exceeds the SQL standard as specified with more features than the widely used products like MySQL and Oracle. The original PostgreSQL was built in 1986 as part of the POSTGRES project at the University of California, Berkeley and has since been the primary database developed in the core product.

Features

Extensive Query Flexibility

- Flexible SQL compliance for advanced users looking for
- Full functionality even when some advanced features are disabled
- Point-in-time recovery and cross-database replication
- Advanced security measures and prepared queries

Advanced Data Type

- Native JSON and JSONB for document-oriented applications with indexing
- Designated integer types for non-standard operations
- Array, range, and zero-valued types
- Custom-defined user-defined aggregates
- C/C++/Haskell, Java, and other object-oriented languages
- Time-series extensions with continuous timelines

Powerful Query Capabilities

- Subqueries across plans and tables
- Common table expressions (CTEs) for recursive queries
- Window functions for analytical solutions
- Materialized views for pre-computed complex results
- Partitioning for managing large datasets
- Parallel query execution for distributed systems

Extensible Architecture

- Customizable extensibility for programming languages (PL/pgSQL, PL/python, PL/etc...)
- Foreign Data Wrappers for connecting to external databases
- Plugable extension architecture
- Over 100 extensions for specialized functionality
- Custom operators and index types

Comprehensive Security

- Secure access for both internal and external control
- Access-based permissions and role isolation
- SSL/TLS encryption for client connections
- Strong authentication or methods including SCRAM and GSSAPI
- Multi-factor for compliance environments

Robust Feature Set

- Stream processing and triggers
- Event triggers for database-level events
- Asynchronous replication with SL-STREAMING
- Distributed transaction support
- Data backup tools for disaster recovery scenarios
- Change data capture for streaming data integration

Stability Options

- High availability for mission-critical workloads
- Consistent performance across different hardware configurations
- Backward compatibility with PostgreSQL
- Shared libraries for efficient resource usage
- High reliability options with custom-built

Performance Optimizations

- Auto-tuning for optimal configuration of queries
- Advanced indexing options (e.g., BRIN, HLL, SP-GIST, etc.)
- Intelligent access paths for maintenance
- Scalable distributed architecture for large-scale planning
- Incremental backups for efficient storage management

Get It Now

Check out our installation guides:

cloudscale

APPUFO

Cloudscale

Elasticsearch

Managed OpenShift

Igeeks

igeeeks

leugo

```
apiVersion: vshn.appcat.vshn.io/v1
kind: VSHNPostgreSQL
metadata:
  name: postgres-appl-prod
spec:
  writeConnectionSecretToRef:
    name: postgres-creds
```

```
kind: Secret
apiVersion: v1
metadata:
  name: postgres-creds
data:
  POSTGRES_URL:
cG9zdGdyZXM6Ly9wb3N0Z3Jlc2piZWlxlWRiMWQtdNDNjOS04YmRAcGdzcWwtYXBwMS1wcm9kLWNsejh0LnZzaG4t
C1hcHAxLXByb2QtY2x6OHQuc3ZjLmNsdXN0ZXIubG9jYWw6NTQzMjMi9wb3N0Z3Jlcw==
  POSTGRES_PASSWORD: YmViMS1kYjYkLTQzYzktOGJk
  POSTGRES_DB: cG9zdGdyZXM=
  POSTGRES_PORT: NTQzMg==
  tls.key:
LS0tLS1CRUdJTiBSU0EgUFJJVkFURSBLRVktLS0tLQpNSU1KS2dJQkFBS0NBZ0VBMHR0YzNjUi9PampmeU83TmpI
1RTaUzPZGhDN1ErRlFHCmVlME43Y3V5R3BNae41RUNpcG1NZFVXeWhBUHBDNDM4dUNNbZrQs2xyTVBsVW5TTDR5U
pYZmFCSkFNUDE3RjlpEetZSFgwb3IxVmZuS1lHTEExqUFpob3NoKytkUm91ZEJXcGpzck14RW05alJGYTQ4Tgp0WH
Ld1NjY2wwSmxHREpseVpWZUczYU04ej14UXFTcmxNZXNqeTdVU1dwVmQ2SUFSCm96VHRrRjBNQ1VuMk9sRXpFTnZ
ZXFCUWR1RFhaV21ZNkhtT2J3TC9BZjFIdVhGb28KS3RQSWtjT1ZxSEppVTJrM29jeEcxYWY5cENJQzV4K20xd1g5
mE0R2NFNXBGKzQ0KwpzcUM2TWRsZ3pLdnNjbHBHULVYmZ22aHpQa0MyC3FRU1VuRmtGaTFFczBhSFNPoVgyVHVH
pYNkVyQStIc3J4dmNWbFVzN0cwZHdrcC9rWHR5UVBDR1F0anJTNUN0TXhTS2UreThnU1VuT2J5SDFLcHYKNCtXWn
vdzJHbkNYS3pvT0kvRUdaME83K0o1MXVrZUJsYWN6dFlmUEU2eEF6RjlzSQo4RFRiWV1Vd095K0M0aUhoZDNPdTR
MmZLdzhWdj12dGFVY1UrTFBGOHc1OTFIWnd1CmtUQk83a2pFdERXZkhRUFFicD1TM1VXV09sVHQ5K2JUcWN3b1lF
3RsNFlrQ0F3RUEKQVFLQ0FnRUF2aC9IMDdzN0VPOTVjVD1LSGVxekk5K1FzMnB1eHVtRmQwZEhRbD1LdDRaVExLY
ZZV21ZeXNabzNLMGxpSVNML2xjTT1iTE9LSudvaWtmVDNUYUtMc0tVOWt5WWVpZzZkvNE1NQ1RxZzVTcnpSbXlaU1
RRTMvekV1cmZzTGtLRmo0NDRvempVMGY0VVNJbUhVdDJ4ZWNBKFEZUmIKME5a5s1pVQ0dzR2M0bDcyMjdpbFduVF
WmdmSkkyNFJRblZid1R4WXpVunNIWGXWAAo1Z1JIdmZ1N2RyP05Vv1ZnQ0I3dU1mbUuVeXlGMLg0Ujhpaxo3Rnd
```




```
apiVersion: vshn.appcat.vshn.io/v1
kind: VSHNPostgreSQL
metadata:
  name: pgsql-app1-prod
spec:
  writeConnectionSecretToRef:
    name: postgres-creds
```



Hide values

POSTGRESQL_DB	postgres
POSTGRESQL_HOST	pgsql-app1-prod-clz8t.vshn-postgresql-pgsql-app1-prod-clz8t.svc.cluster.local
POSTGRESQL_PASSWORD	beb1-db1d-43c9-8bd
POSTGRESQL_PORT	5432
POSTGRESQL_URL	postgres://postgres:beb1-db1d-43c9-8bd@pgsql-app1-prod-clz8t.vshn-postgresql-pgsql-app1-prod-clz8t.svc.cluster.local:5432/postgres
POSTGRESQL_USER	postgres
ca.crt	-----BEGIN CERTIFICATE----- MIIHjCCA26gAwIBAgIRAM1jdZXzrbjryEg00aj5v5MwDQYJKoZIhvcNAQELBQAw FjEUMBIGA1UEChMLdnNobi1hcHBjYXQwHhcNMjQwNjEyMTM2WmcNMzQwNjEw MjEwMTM2WjAwMRQwEgYDVQKKEwt2c2huLWFwcGNhdDCCAiIwDQYJKoZIhvcNAQEB BQADggIPADCCAgoCggIBANLbXN3Efo438juzYxymUYrxBK+7XOCcD/k0ahTnYQu



AppCat

AppCat / PostgreSQL / On cloudscale.ch / Creation

» The Application Catalog

▼ PostgreSQL

» On Exoscale

▼ On cloudscale.ch

Creation

Usage

Connect

Deletion

Backup

Restore

Security

Scheduled Maintenance

Plans and Sizing

Service Level

Agreement

Database Deletion

Protection

Encrypted Storage

User Alerting

Instance Update

Strategy

PostgreSQL Extensions

Replication and High
AvailabilityManage Users and
Databases

AppCat

default ▼

Create a PostgreSQL service

Apply the following object on your namespace, as specified by its YAML description.

Example to create a PostgreSQL instance

```
apiVersion: vshn.appcat.vshn.io/v1
kind: VSHNPostgreSQL
metadata:
  name: pgsql-app1-prod ①
  namespace: prod-app ②
spec:
  parameters:
    service:
      majorVersion: "15" ③
      pgSettings:
        timezone: Europe/Zurich ④
  size: ⑤
  plan: plus-4
  backup: ⑥
    schedule: "30 23 * * *"
    retention: 12
  writeConnectionSecretToRef:
    name: postgres-creds ⑦
```

① Instance name

② The namespace where the object will be created

③ PostgreSQL version

④ Specify custom PostgreSQL settings [optional]

⑤ Size of the PostgreSQL instance. See [Plans and Sizing](#) for more information.

⑥ Backup configuration, `schedule` standard cron: en.wikipedia.org/wiki/Cron, `retention` field specify how many backups should be kept

⑦ Secret where the connection details are provisioned. This secret shouldn't exist before creation.

1

To get more information about all available configuration options, please see the [API Reference](#)

Servala frontend



PostgreSQL at Hetzner Cloud

PostgreSQL at Hetzner Cloud
Database

Service Provider Zone *

[General](#)
[Parameters](#)
[Publish connection details to](#)
[Others](#)

Instances

Instances configures the number of PostgreSQL instances for the cluster. Each instance contains one Postgres server. Out of all Postgres servers, one is elected as the primary, the rest remain as read-only replicas.

Encryption: Enabled

Enabled specifies if the instance should use encrypted storage for the instance.

Replication: Mode

Mode defines the replication mode applied to the whole cluster. Possible values are: "async"(default), "sync", and "strict-sync" "async". When in asynchronous mode the cluster is allowed to lose some committed transactions. When the primary server fails or becomes unavailable for any other reason a sufficiently healthy standby will automatically be promoted to primary. Any transactions that have not been replicated to that standby remain in a "forked timeline" on the primary, and are effectively unrecoverable "sync". When in synchronous mode a standby will not be promoted unless it is certain that the standby contains all transactions that may have returned a successful commit status to client. This means that the system may be unavailable for writes even though some servers are available. "strict-sync": When it is absolutely necessary to guarantee that each write is stored durably on at least two nodes, use the strict synchronous mode. This mode prevents synchronous replication to be switched off on the primary when no synchronous standby candidates are available. As a downside, the primary will not be available for writes, blocking all client write requests until at least one synchronous replica comes up. NOTE: We recommend to always use three instances when setting the mode to "strict-sync".

pgtest

[Edit](#) [Delete](#)

Details

Service PostgreSQL
Service Provider Hetzner Cloud
Control Plane Nuremberg (nbg1-dc3)
Created By Tobias Brunner
Created At 06/23/2025 9:55 a.m.
Updated At 06/23/2025 9:55 a.m.
Status Active

Status

Type	Status	Last Transition Time	Reason	Message
Synced	True		ReconcileSuccess	-
Ready	True		Available	-

Specification

[General](#)
[Parameters](#)

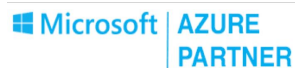
Composite delete policy Background
Composition update policy Automatic
Composition ref: Name vshnpostgres.vshn.appcat.vshn.io
Composition revision ref: Name vshnpostgres.vshn.appcat.vshn.io-137abe4

Connection Credentials

Name	Value
------	-------

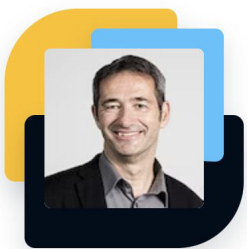
Public and private cloud partners

Public cloud:



Enterprise private cloud:





Daniel Bernasconi

CTO and Member of
the Executive Board
of Finnova

"I would recommend VSHN to all who would like to have a 'platform as a service' model that is not restricted to a single infrastructure provider. Those looking for a reliable Swiss partner who is also prepared to talk about individual requirements are also well served."

finnova.

finnova AG Bankware

Challenge

- Need for a stable and reliable platform to run applications and other software and support automated deployment pipeline.
- Lack of internal expertise in operating a secure & scalable platform like OpenShift.
- Swiss regulatory framework and compliance requirements.

Solution

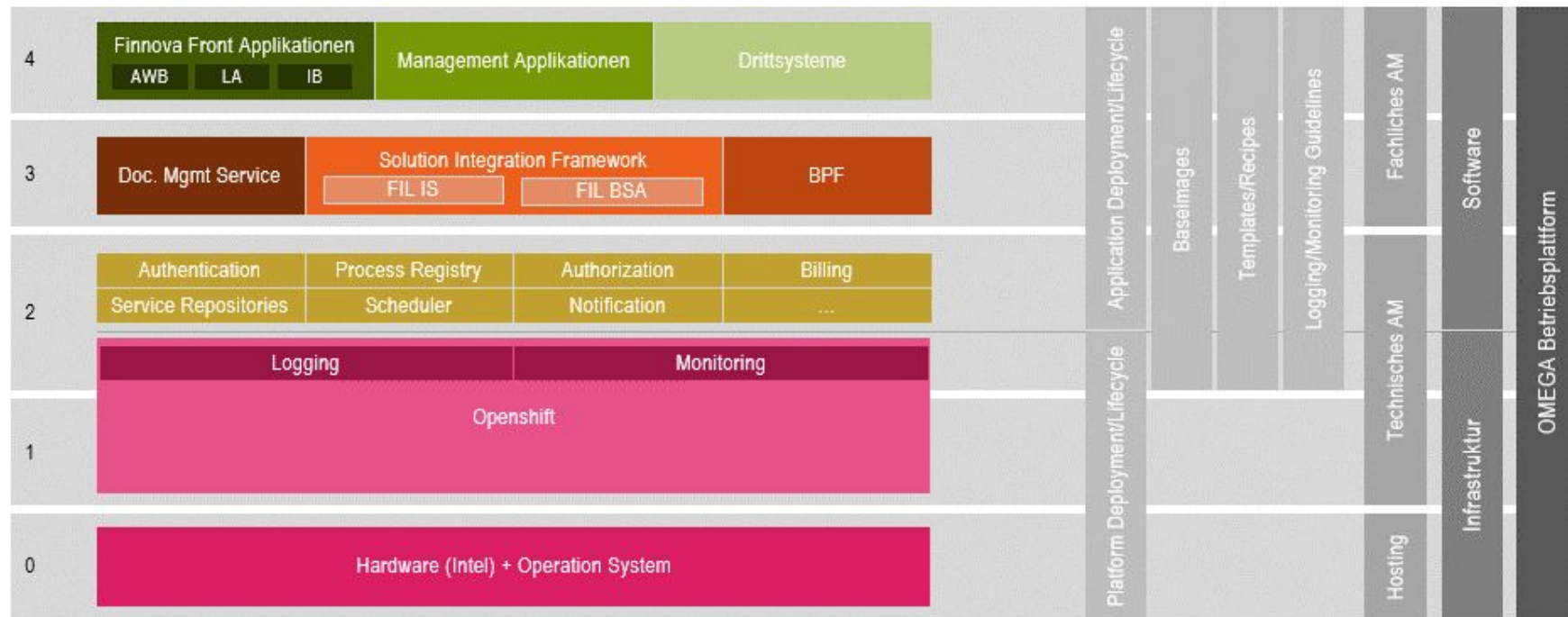
- APPUIO Managed OpenShift Clusters with Premium support
- Managed GitLab
- Custom Extended Support and additional managed services like Postgres, MongoDB.

Benefits

- The entire Finnova Omega Platform operates on OpenShift and scaling seamlessly as the business grows.
- **Enterprise-Grade Security**
- **Faster Time-to-Market** by automating manual, time-consuming tasks, such as building fleets of new databases on new hardware with the push of a button.
- Finnova **reduced its operational burden** and realized substantial cost savings.

RUNTIME ENVIRONMENT

OMEGA-Betriebsplattform



acrevis Bank

- VSHN: OpenShift, middleware, WAF, developer platform
- Puzzle: “Lagoon”: development of frontend & backend applications, pipelines, integration layer towards core banking, IAM
- Since 2019



Lagoon bietet Betriebssicherheit und Flexibilität zu einem fairen Preis. Dank dieser Lösung können wir sehr schnell und unkompliziert auf die Bedürfnisse unserer Kunden reagieren und moderne Lösungen anbieten.»

Christian Gentsch,
Mitglied der Geschäftsleitung
acrevi Bank AG und CEO Finanz-
Logistik AG

Lagoon ist bei der acretis Bank AG seit September 2019 in Betrieb.

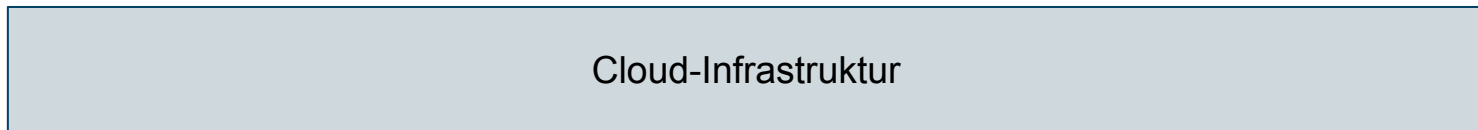
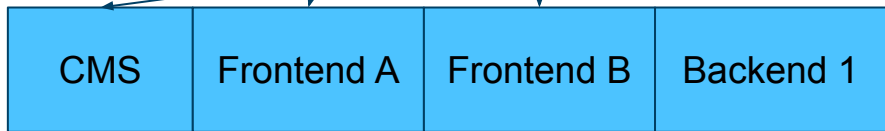
acreviis lagoon platform

Kunde

Berater

Supporter

Entwickler



IAM
Corebanking
DMS

Taurus expands globally in partnership with Red Hat and VSHN

TAURUS

Software and services

Red Hat® OpenShift®

Partner

VSHN

Since its inception in Switzerland in 2018, Taurus has grown rapidly to become the leading digital asset infrastructure provider, allowing financial institutions to issue, book, transfer, and trade any digital asset within 1 platform. When vanilla Kubernetes became too time-consuming to manage, OKD (the community version of OpenShift) provided the answer. As customer numbers grew, Taurus migrated its platform to Red Hat OpenShift managed by Red Hat partner, VSHN. The Red Hat technology provides stability, reliability, and scalability for a platform relied upon by a growing number of financial institutions. VSHN ensures the platform continues to run smoothly and reliably as Taurus expands globally.



Technology, fintech

81+ employees

Benefits

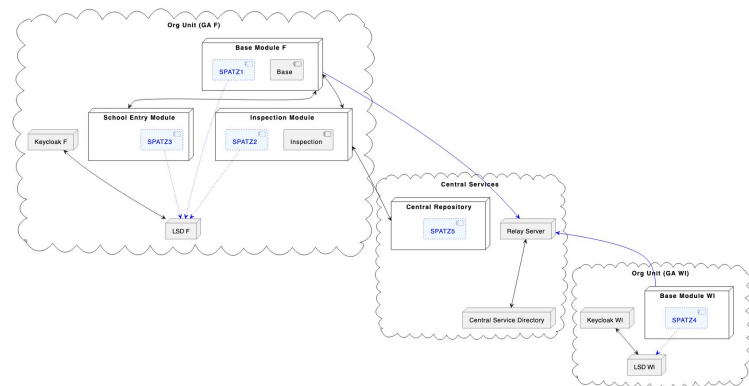
- ▶ Cut upgrade downtime to zero while maximizing reliability
- ▶ Brought peace of mind to Taurus and its customers
- ▶ Met customer SLAs 100% of the time

“We wanted a Red Hat partner to act as a single point of contact for our Red Hat technology. We had worked with VSHN before and had full trust in them.”

Sébastien Pasche
VP of Engineering, Taurus S.A.

Health Department of Frankfurt aM, DE

- Health authority case processing software in open source
 - <https://gitlab.opencode.de/ga-lotse/>
 - General modules such as user/login, calendar, statistics, open data, etc
 - Subject-specific modules such as enrollment examinations, hygiene checks, etc
- Open standards for operation and data storage, each office has its (customizable) instance, remains the owner of its (personal) data
- VSHN won the public tender to operate OpenShift and services like databases, etc in Germany



Health Info Net HIN.ch

- Service provider for Swiss healthcare, like encrypted @hin.ch email
- cultural shift from single-vendor, manual, physical waterfall to dual-vendor, self-service automated, VPC-based DevOps model
- Solution: multi-cloud container platform, GitOps, catalog of platform engineering services for teams & vendors



Conclusion

- Kubernetes operator pattern enables decentralization
- Decentralization enables scale
- I need to master each technology I use - or partner with somebody who does
- Automation enables self-service and compliance

Thank you and feedback very welcome!



Aarno Aukia

@aarnoaukia a@vs.hn

<https://www.linkedin.com/in/aukia/>

